

Transport Layer Overview

Kai Shen

10/12/2009 CSC 257/457 - Fall 2009 1

Transport-layer Overview

- **Network layer:** host-to-host logical communication between *hosts*.
- **Transport layer:** logical communication between *applications*.
 - multiple comm. applications can reside in one host.
 - comm. applications can be a Web browser/server, a FTP client/server, etc.

10/12/2009 CSC 257/457 - Fall 2009 2

Multiplexing/demultiplexing

○ = comm. application □ = socket

P1 on host A is communicating with P3 on host B; while at the same time, P2 on host A is communicating with P4 on host B.

10/12/2009 CSC 257/457 - Fall 2009 3

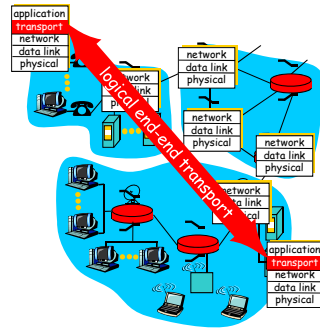
Transport-layer Service Model

- **Transport layer:** logical communication between *applications*.
 - multiplexing/demultiplexing
- Possible additional services:
 - **reliable data transfer** (guaranteed arrival, no error, in-order)
 - **flow control** (keep sender from overrunning receiver): good for myself
 - **congestion control** (keep sender from overrunning network): good for the community
 - **delay/bandwidth guarantee**
 -

10/12/2009 CSC 257/457 - Fall 2009 4

Transport-layer in the Internet Architecture

- End-host only transport implementation
- Simple network layer, complex transport
- Benefits:
 - Fast, stable network layer
 - Flexible transport layer
- Challenge:
 - Transport functions without router assistance



10/12/2009

CSC 257/457 - Fall 2009

5

Internet Transport-layer Protocols

- UDP: minimum service transport
 - multiplexing/demultiplexing
 - error detection
- TCP: full service transport
 - multiplexing/demultiplexing
 - error detection
 - reliable data transfer
 - flow control
 - congestion control
- services not available:
 - delay guarantees
 - bandwidth guarantees
 - why?

10/12/2009

CSC 257/457 - Fall 2009

6

Connection-oriented / Connectionless Transport

- What is a *connection*?
 - logical grouping of data segments for a comm. session between the same pair of end-points
 - state maintained for such session
- Why is a connection useful?
 - implement lost data detection and retransmission
 - track network performance using a history of performance measurement
 -

10/12/2009

CSC 257/457 - Fall 2009

7

Connection-oriented / Connectionless Transport

- UDP is *connectionless*:
 - each UDP segment handled independently of others
 - minimum service
- TCP is *connection-oriented*:
 - state information for each TCP connection:
 - buffer of recently transmitted segments
 - performance of recently received segments
 - statistics about the other side
 - additional costs:
 - connection initialization/teardown required at the beginning/end of communications
 - additional space consumption during the communications

10/12/2009

CSC 257/457 - Fall 2009

8

Outline

- Overview of transport-layer services/protocols
- Multiplexing and demultiplexing**
- Connectionless transport: UDP
- Principles of reliable data transfer
- Connection-oriented transport: TCP
 - reliable data transfer
 - flow control
 - connection management
- Principles of congestion control
- Congestion control in TCP

10/12/2009 CSC 257/457 - Fall 2009 9

Multiplexing/demultiplexing

host A host B

○ = comm. application □ = socket

P1 on host A is communicating with P3 on host B; while at the same time, P2 on host A is communicating with P4 on host B.

10/12/2009 CSC 257/457 - Fall 2009 10

How multiplexing/demultiplexing works?

IP header	TCP/UDP header	application data
-----------	----------------	------------------

IP packet format

- using port numbers**
 - each IP packet has source IP address, destination IP address
 - each IP packet carries a transport-layer segment
 - each segment has source, destination port number
- dest. IP address** for routing to the host; **IP addresses and port numbers** for going to appropriate socket in the dest. host.

10/12/2009 CSC 257/457 - Fall 2009 11

UDP Demultiplexing: An Example

UDP socket identified by dest. IP address and port number

Machine A Machine C Machine B

port 1234 port 5678

SP: XXXX DP: 1234

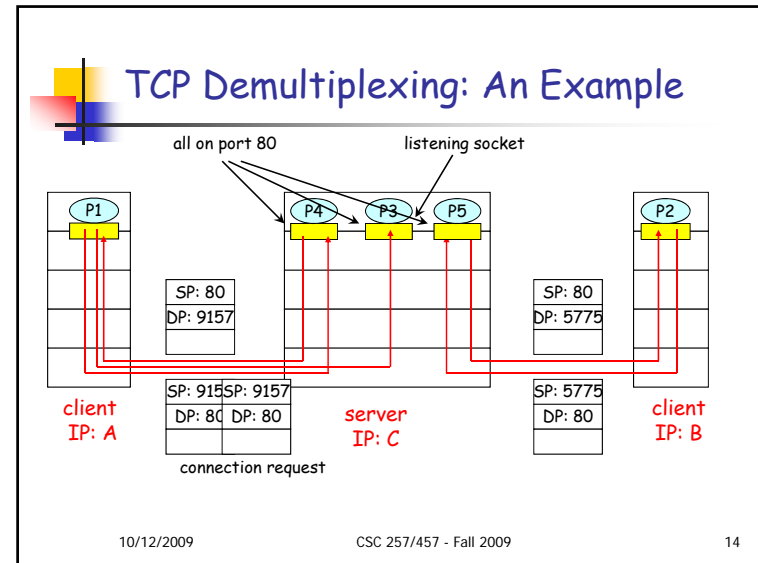
SP: YYY DP: 5678

10/12/2009 CSC 257/457 - Fall 2009 12

TCP Demultiplexing (Connection-oriented)

- Multiple TCP connections may simultaneously share a single port:
 - **Example:** web server (on default port 80) has multiple connections, one for each client
- TCP socket identified by 4-tuple:
 - (source IP address, source port number, dest IP address, dest port number)
 - recv host uses all four values to direct segment to appropriate socket

10/12/2009
CSC 257/457 - Fall 2009
13



UDP: User Datagram Protocol

- UDP is connectionless
- UDP does
 - multiplexing/demultiplexing
 - simple error detection
- UDP does not do
 - reliable data transfer, flow control, congestion control ...

← 32 bits →	
source port #	dest port #
length	checksum
Application data (message)	

Length, in bytes of UDP segment, including header

UDP segment format

10/12/2009
CSC 257/457 - Fall 2009
15

UDP Checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segments

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
 - NO: error detected
 - YES: no error detected.

Why are there error detections in transport, network, and link layers?

10/12/2009
CSC 257/457 - Fall 2009
16

What is good about UDP?

- less overhead:
 - no connection establishment (which can add delay)
 - small segment header
 - no space cost for connection state
- more secure: less susceptible to DoS attacks
- flexibility in doing things your own way
 - no congestion control: UDP can blast away as fast as desired (may not be good for others, but not bad for my own connection ☺)
- simple ⇒ more robust

10/12/2009

CSC 257/457 - Fall 2009

17

UDP In Practice

- When to use UDP?
 - TCP services are not useful, or even hurtful
 - Useful, but in customized ways
- Used for:
 - routing exchanges between routers
 - network performance measurements
 - streaming multimedia apps
 - loss tolerant, rate sensitive
 - flow control and loss recovery sometimes implemented at the application level

10/12/2009

CSC 257/457 - Fall 2009

18

Disclaimer

- Parts of the lecture slides contain original work of James Kurose, Larry Peterson, and Keith Ross. The slides are intended for the sole purpose of instruction of computer networks at the University of Rochester. All copyrighted materials belong to their original owner(s).

10/12/2009

CSC 257/457 - Fall 2009

19